Amendments to the Claims

Please amend the claims to read as follows.

1-2. (Canceled.)

3. (Currently Amended): A computer-implemented method of memory management, comprising the steps of:

providing a smart pointer for association with a memory-resident element, the smart pointer including a next pointer;

providing an assignment means for assigning the next pointer to point to the smart pointer thereby creating a linked list comprising the smart pointer; ~~and~~

providing a comparison means for comparing the value of the next pointer to the value of the memory location of the smart pointer in which the ~~selected~~ next pointer is included, whereby a determination can be made if the ~~ring~~ <u>linked list</u> contains more than one smart pointer<u>; and</u>

<u>deleting the memory-resident element associated with the smart pointer if the value of the next pointer of the smart pointer is equal to the value of the memory location of the smart pointer in which the next pointer is included and not deleting the memory-resident element if the value of the next pointer of the smart pointer is not equal to the value of the memory location of the smart pointer in which the next pointer is included.</u>

4. (Original) The method according to claim 3, wherein the method comprises the step of providing a common base to the smart pointer.

5. (Original) The method according to claim 3, wherein the element is an object in an object-oriented programming environment.

6. (Original) The method according to claim 5, wherein the smart pointer includes an object pointer for pointing to the object.

7. (Original) The method according to claim 3, wherein the linked list comprises a ring.

8. (Original) The method according to claim 3, wherein the smart pointer includes a previous pointer.

9. (Original) The method according to claim 8, comprising a step of providing an assignment means for assigning the previous pointer to point to the smart pointer, thereby creating a bi-directional, doubly-linked list.

10. (Original) The method according to claim 9, wherein the linked list comprises a ring.

11. (Canceled.)

12. (Original) The method according to claim 3, wherein the smart pointer includes a first smart pointer, and wherein the method comprises the step of providing an attachment means for attaching a second smart pointer associated with the memory-resident element to the linked list.

13. (Currently Amended): A computer-implemented method of memory management, comprising the steps of:

        providing a linked list comprising a smart pointer associated with a memory-resident element, the smart pointer including a next-pointer for pointing to the smart pointer; and

        providing a comparison means for comparing the value of the memory location of the smart pointer to the value of the next-pointer of the smart pointer, to provide a determination whether the linked list contains only the smart pointer; and

deleting the memory-resident element associated with the smart pointer if the value of the next-pointer of the smart pointer is equal to the value of the memory location of the smart pointer in which the next-pointer is included and not deleting the memory-resident element if the value of the next-pointer of the smart pointer is not equal to the value of the memory location of the smart pointer in which the next-pointer is included.

14. (Canceled.)

15. (Original) The method according to claim 13, wherein the element is an object in an object-oriented programming environment.

16. (Original) The method according to claim 15, wherein the smart pointer includes an object pointer for pointing to the object.

17. (Original) The method according to claim 13, wherein the linked list comprises a ring.

18. (Original) The method according to claim 13, wherein the smart pointer includes a previous pointer.

19. (Original) The method according to claim 18, comprising a step of providing an assignment means for assigning the previous pointer to point to the smart pointer, thereby creating a bi-directional, doubly-linked list.

20. (Original) The method according to claim 19, wherein the linked list comprises a ring.

21. (Original) The method according to claim 13, comprising a step of providing a deletion means for deleting the memory-resident element associated with the smart pointer if

the value of the next-pointer of the smart pointer is equal to the value of the memory location of the smart pointer in which the next-pointer is included.

22. (Original) The method according to claim 13, wherein the smart pointer includes a first smart pointer, and wherein the method comprises the step of providing an attachment means for attaching a second smart pointer associated with the memory-resident element to the linked list.

23. (Currently Amended): A computer-implemented method of memory management, comprising the steps of:

       providing a linked list comprising a first smart pointer and a second smart pointer each associated with a memory-resident element, the first smart pointer including a first next-pointer for pointing to the second smart pointer and the second smart pointer including a second next-pointer for pointing to the first smart pointer; and

       providing a comparison means for comparing the value of the memory location of a selected smart pointer giving up its association with the memory-resident element to the value of the next-pointer of the selected smart pointer, to provide a determination whether the linked list contains only the selected smart pointer; and

       providing a deletion means for deleting the memory-resident element associated with the smart pointer if the value of the next-pointer of the selected smart pointer is equal to the value of the memory location of the selected smart pointer in which the next-pointer is included and not deleting the memory-resident element if the value of the next-pointer of the selected smart pointer is not equal to the value of the memory location of the selected smart pointer in which the next pointer is included.

24. (Canceled.)

25. (Original) The method according to claim 23, wherein the linked list comprises a ring.

Reply under 37 CFR 1.116 – Expedited Procedure - Technology Center 2124
Application No.: 10/079,928   **Art Unit: 2124**
Docket No.: 3434-P02437US1   **Examiner: Satish Rampuria**

26. (Original) The method according to claim 23, wherein the first smart pointer and the second smart pointer each include a previous pointer.

27. (Original) The method according to claim 26, comprising a step of providing an assignment means for assigning the previous pointer of the first smart pointer to point to the second smart pointer and for assigning the previous pointer of the second smart pointer to point to the first smart pointer, thereby creating a bi-directional, doubly-linked list.

28. (Original) The method according to claim 27, wherein the linked list comprises a ring.

29. (Canceled.)

30. (Original) The method according to claim 23, comprising the step of providing an attachment means for attaching a third smart pointer associated with the memory-resident element to the linked list.

31. (Original) The method according to claim 23, comprising the step of providing a common base to the smart pointers.

32. (Original) The method according to claim 23, wherein the element is an object in an object-oriented programming environment.

33. (Original) The method according to claim 32, wherein the first smart pointer and the second smart pointer each include an object pointer for pointing to the object.

34. (Original) The method according to claim 32, wherein the first smart pointer is associated with a first object of a first class and the second smart pointer is associated with a second object of a second class, and wherein the method comprises the step of providing a

conversion means for providing automatic conversion between the first smart pointer and the second smart pointer.

35. (Currently Amended): A computer-implemented method of memory management, comprising the steps of:

providing a smart pointer for association with a memory-resident element, the smart pointer including a previous pointer;

providing an assignment means for assigning the previous pointer to point to the smart pointer thereby creating a linked list comprising the smart pointer; ~~and~~

providing a comparison means for comparing the value of the previous pointer to the value of the memory location of the smart pointer in which the selected previous pointer is included, whereby a determination can be made if the ring contains more than one smart pointer; and

deleting the memory-resident element associated with the smart pointer if the value of the previous pointer of the smart pointer is equal to the value of the memory location of the smart pointer in which the previous pointer is included and not deleting the memory-resident element if the value of the previous pointer of the smart pointer is not equal to the value of the memory location of the smart pointer in which the previous pointer is included.